Ontology reviewed with a Case Study
and discussed about
the development approaches, their adaptation
and
applications.

February 2023

Version History:

No.	Date	Comments
1.0	26-Jan-2023	Outlined the structure
1.1	13-Feb-2023	Review and applications

1 Table of Contents

2 Why Ontology? 3

3 Case Review: "Ontology development for agriculture domain" 3

4 Further Applications 4

Software Engineering Process 4

5 Rationale **Error! Bookmark not defined.**

6 Pros & Cons **Error! Bookmark not defined.**

7 Analysis of Pros & Cons **Error! Bookmark not defined.**

8 Demonstration of the Approaches **Error! Bookmark not defined.**

9 Conclusion 6

10 References 6

2 Why Ontology?

We have lots of knowledge in the form of unstructured data from various domains and sources. Building a formal description as a standard is necessary to translate the relationship between knowledge and the domain.

It simply means clarifying how the relationship is understood for synonyms, antonyms, part of a whole, cause and effect and item in a category. This is what we know as an analogy as well.

So, to make sure that the representation of knowledge can be shared, re-used and build knowledge about the domain, all the descriptions and unstructured pieces of information should be understood by formal specifications like instance (of objects), classes, attributes, their relationships, constraints, rules and principles or axiom.

Ontologies have become a part of the W3C standards as a building block to Semantic Technology by providing a required structure to stitch information with other pieces of data on the Web.

Large organisations benefit by managing their data that has value (understood) and quality due to the inherent ontology characteristic of ensuring the interconnectedness and interoperability of the model.

The Web Ontology Language (OWL) was designed as the semantic web computational logic-based language to represent the relationship between complex knowledge and how they are related.

3 Case Review: "Ontology development for agriculture domain"

The paper, *Malik, N., Sharan, A. & Hijam, D. (2015) March. Ontology development for agriculture domain. In 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 738-742). IEEE,* is being reviewed to understand why Ontology was employed as mean for addressing complexities in the agriculture domain,

The objective of the work was to develop an Ontology for the agriculture domain by consultation with the domain experts and understanding from the knowledgebase, resources like Agrovoc, AGRIS, Agropedia and Agro tagger. Within the resources, limitations were found, for example, Agrovoc, like relationship inconsistencies and thus the requirement to develop an ontology for this domain to represent a structured knowledge.

The ontology development was initiated by classifying five important classes – plants, diseases, pests, pesticides and fertilisers. **Protégé,** as the open-source tool, was used for ontology development.

The work comprised in
- building or development of the ontology
- classification or snapshots

The development of Ontology was based on by representing the tuple, represented as O = < **C***oncepts*, **R***elations*, **I***nstances*, **A***xioms*> where the specification is explicit to considering classes, attributes, relationships and instances.

To elaborate O = <C, R, I, A>,
C = set of classes (concepts of the domain)
R = set of relationships between classes in C
I = set of class instances where I is linked by R
A = set of Axioms
([1]Scharffe, F, 2009; [2] Meilicke, C, 2011)

Since there are different ways in modelling a domain ontology, the solution built for this work was based on "anticipation and extensions in mind of the developer". This is because the development process needs to be an iterative process by making sure that the concepts are represented correctly that describe the domain.

A seven step process was proposed to develop:
Step 1. Determine the domain and scope of the ontology
Step 2. Consider reusing existing ontologies
Step 3. Enumerate important terms in the ontology
Step 4. Define the classes and the class hierarchy
Step 5. Define the properties of classes—slots
Step 6. Define the facets of the slots
Step 7. Create instances

This lead to the building of Snapshots with identification of top five classes - Plant, Disease, Pest, Pesticide and Fertilizer. These were then further created with subclasses and their properties

By doing this, a single ontology was achieved with a generic approach that has advantages of being an enabler to structure more knowledge and relations. Since their work was done from scratch, it has less ambiguity and room for more enhancements and inputs.

4 Further Applications

- ### Software Engineering Process

With the onset of pandemic (COVID), software development and management has seen a wave in the increase of virtualisation of teams. The SDLC (Software Development Life Cycle) has risen to be more virtually collaborative with challenges towards operations and support ([3] K. S. Kaswan, S. Choudhary and K. Sharma, 2015).

Improvements have been realised with researches from the Semantic Web Technology (SW) and Software Engineering that with the help of Ontologies and SW techniques like RDF, RDFS, SPARQL, etc. leads to improvements in the SDLC or the engineering processes ([4] R. Studer, R. Benjamins and D. Fensel, 1998).

Though this brought an enhancement in the process but still posses with challenges and problems. One of them is to identify ambiguities in the Software Requirement Specifications (SRS) that is inception for building the designs, architecture, test plans, etc.

By using Ontology as a basis for software development, it can be used to capture scopes and requirements with description about relationships ([5] Y Zhao, J Dong and T Peng, 2009).

A proposed framework was implemented to detect the ambiguities within the SRS by using application domain ontology with the help of Natural Language Processing (NLP) and Semantic Web Techniques ([6] Bhatia, M. P. S., Akshi Kumar, and Rohit Beniwal, 2016).

The approach and framework used to understand the ambiguities (lexical, syntactic, semantic, pragmatic, vagueness, language, etc.) that could be present in the SRS document. The lexical and syntactic could be detected by using NLP techniques, but it requires the other ambiguities need to be improved.

The framework was split into phases of - Development and Reasoning. This was done so that with the development phase, the domain ontology could be built to cover the SRS into a formal specification. This becomes the semantic reasoner to infer the logical understanding from the axioms. For implementations, the ontology editor tools were used and information about reasoners in OWL w3 site was available.

Though this removes the ambiguities, more work is still required to build this within the agile tools like JIRA, Confluence, etc. that are used to manage most of the engineering processes.

○ E-learning recommender

E-learning has become a known area with wide audiences like students (primary, high or college and university), company and employees, technicians or engineers, etc. This onset has led to a load of information in various channels like google, white-papers, youtube, udemy, linked-in, etc. So, it becomes difficult to recommend the resource to the learner from the sea of resources.

To overcome this an ontology can be built that would be used to provide the knowledge about the learner and the resources.

A comprehensive study was made by Tarus, John K., Zhendong Niu, and Ghulam Mustafa in 2018 in this area to define a recommender system for e-learning by using ontology domain knowledge.

The work was based on by

- Summarising the data (recommendation techniques) and graphically representing between years 2005 and 2014
- Analysing and categorising of the ontology types and representation languages
- Future trends and next steps for e-learner recommenders

A taxonomy for the recommendation techniques was built

collaborative filtering (CF) - users with similar tastes
content-based (CB) - similar in content features
knowledge-based (KB) - relation between users and the item
demographic-based (DB) - hybrid recommendation
utility-based (UB) - hybrid recommendation
context-aware (CA) - contextual information
trust-aware based (TA) - future actions

fuzzy-based (FB) - uncertain or vague information
social network-based (SB) - profiles and user relationship
group-based (GB) - group of users
hybrid recommendation - hybrid of two or more techniques

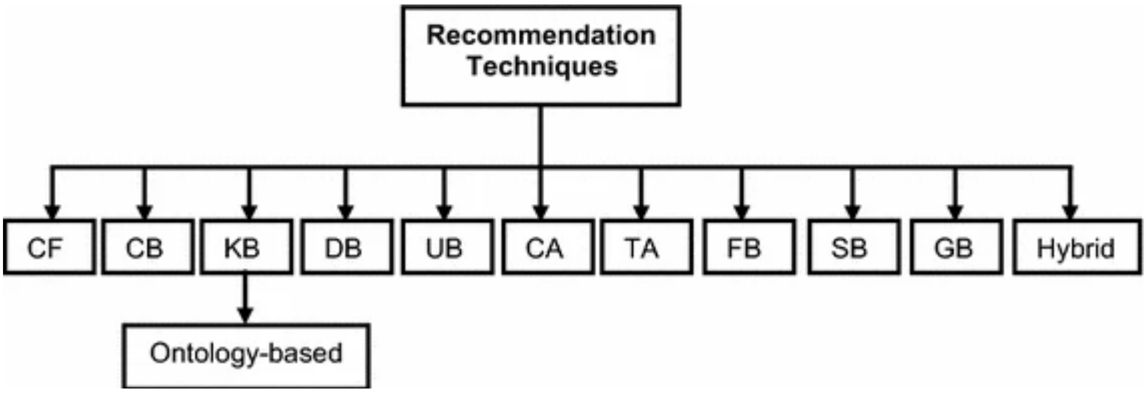

Figure 1: Recommendation techniques (Tarus, John K., Zhendong Niu, and Ghulam Mustafa, 2018)

Further on this, the publications were put into qualitative analysis and categorisations relevant to the recommendation techniques. The information was then extracted and reviewed for any mismatches or disagreements.

There is room for further improvements within the e-learning to make it a more personalised knowledge-based recommender.

5 Conclusion

More and more applications are being designed that will be ontology based, with the relevant data available in almost all domains. Some data are readily accessible, but some are not. The data is required to become more knowledge-based and pertinent to users. Though the ontology models are becoming the building blocks but there still needs to be ontology knowledge models required that remove spurious or spam, or incorrect pieces of information.

The visualisation of Ontology with Protege tools is enhanced as a concept to see knowledge in a machine-readable format. The popularity of ontology models for knowledge management services and applications has given way to visualise in a better state. Tools focussing on making this to be available graphically are helping towards assessments and analysis.

6 References

- [1] Scharffe, F.: Correspondence Patterns Representation. PhD thesis, University of Innsbruck (2009)
- [2] Meilicke, C.: Alignment Incoherence in Ontology Matching. PhD thesis Universitat Mannheim, Mannheim (2011)
- [3] K. S. Kaswan, S. Choudhary and K. Sharma, "Software Reliability Modeling using Soft Computing Techniques: Critical Review", J Inform Tech SoftwEng, vol. 5, no. 144, 2015.
- [4] R. Studer, R. Benjamins and D. Fensel, "Knowledge engineering: Principles and methods", Data & Knowledge Engineering, vol. 25, no. 1, pp. 161-198, 1998.

- [5] Y Zhao, J Dong and T Peng, "Ontology classification for semantic-web-based software engineering Services Computing IEEE Transactions on Services Computing", vol. 2, no. 4, pp. 303-317, 2009
- [6] Bhatia, M. P. S., Akshi Kumar, and Rohit Beniwal. "Ontology based framework for detecting ambiguities in software requirements specification." 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom). IEEE, 2016.
- [7] Tarus, John K., Zhendong Niu, and Ghulam Mustafa. "Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning." Artificial intelligence review 50 (2018): 21-48.